On Tevatron TuneTracker,

Status Report & Plans

Paul Lebrun

Fermilab

January 30 2003

The team…

- *Jim Patrick, Charlie Briegel, Ron Rechenmaker* for Control and D.A. software
- *Dean Still, Charlie Briegel* HP3561a installation & support, access to VSA tune data.
- *John Marraffino*, offline software and ROOT interfacing
- *Vladimir Shiltsev, & TeV dept*, for their support and patience in MCR..

Outline

– Goal and Scope of this project.
– Status, before Shutdown January 03
– Brief description Algorithm used in fitting, and C++/Java implementation.
– Examples of fits
– Prospects: What's next, a plan for FY03-04
  • One month Horizon (March 1)
  • 3 month (May 1)
  • Later

– *Note : This document is based on two previous talks written in December-02 and January 03 Those, as well as this document, are stored in document number 299 at http://beamdocs.fnal.gov*

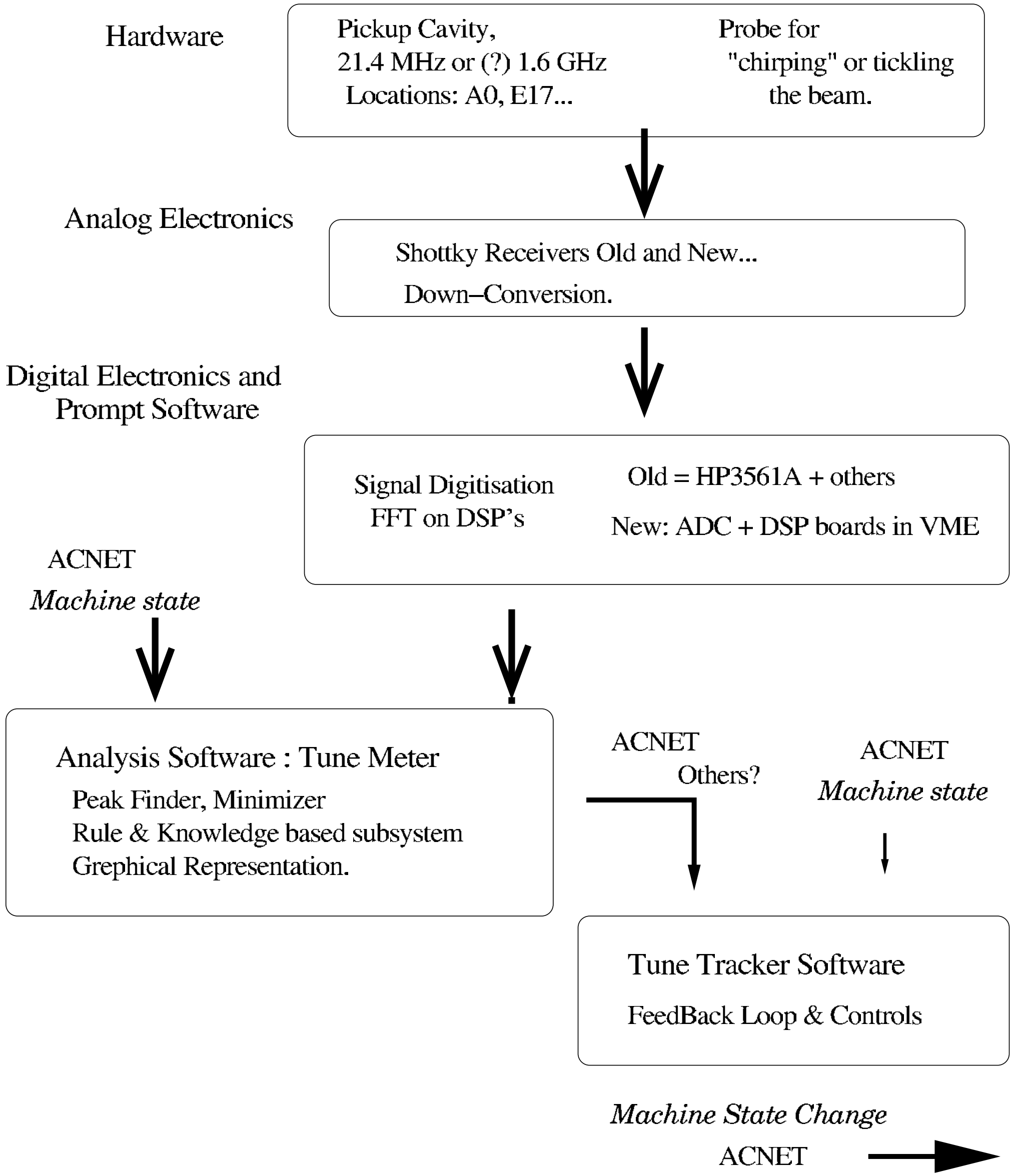
Tevatron Tune Tracking: Goal & Scope

- Automatic fits of the Tune Spectrum Analyzer data seems a difficult task, as it is just a mess of broad bump, narrow signals, and "mostly noise" (especially for coalesced beams)
- Goal of a Tune Meter : express "the art of picking the right line" into a reproducible algorithm that can be implemented on a modern computer, and can be run at ~ 1 Hz.
  - To improve the overall reliability of such measurements.
  - Reduce clock time to doing such measurements
  - Allow the implementation a tune tracker, based a straight feedback loop using this tune meter.
- Scope:
  - Short term: Using existing equipment, (21.4 MHz Shottky, HP3561a) and new software (C++, Java, Root,..)
  - Long Term: dedicated Front-end subsystem with better digitization and FFT on DSP, refine analysis software…

Tevatron Tune Tracking: Ultimate Goal

– A Tune Tracker will possibly allow us to reduce the store turn-around:
  - By automating tunes, Chromaticity and coupling measurements
  - May be, skipping some of the ramp up-down hysteris cycles, as we will be able to track/correct the tune on the fly..
– Automated "parsing" of some ramp, and tune/Chrom. Corrections. (Such ramp will probably take a bit longer than 85 seconds, it may be worth considering).
– ➔ improved integrated luminosity. Hard to quantify, but most likely real..

Tune Meter/Tracker : A simplified "System" View

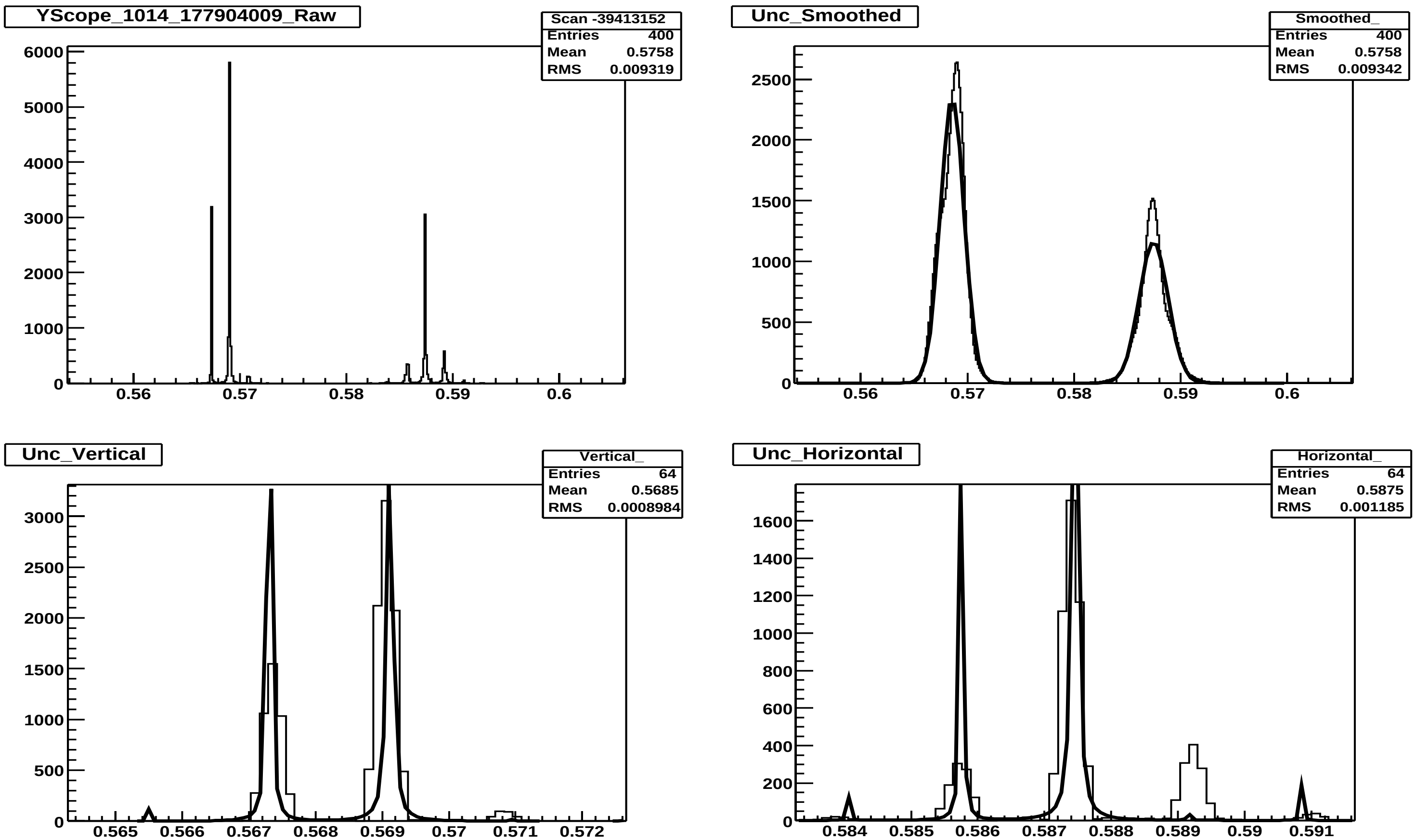


Brief Status and History.

- The vsamcr files (from the C44 page) have been analyzed by John Marraffino, using a C++ root based fitting program, showing that some information could be gained.
- An HP3561A “box” has been connected to the existing proton shottky signal by Dean Still and Charlie Briegel…
- Who has also written a nice ACNET Read Wave Form utility..
- Which, I am able to read on the development system “nova.fnal.gov”
- And fit, using the infrastructure written by John M., based on the root package.
- And, thanks to a XML-RPC based library written in collaboration BD/CDF, we are now writing the result of the fits to ACNET
- Which are “datalogged” on node “Inst2” and the D44 1Hz Archiver.

Uncoalesced Beam, taken during Mike Martens Tev. Tune studies, Dec 11 2002.

16:36

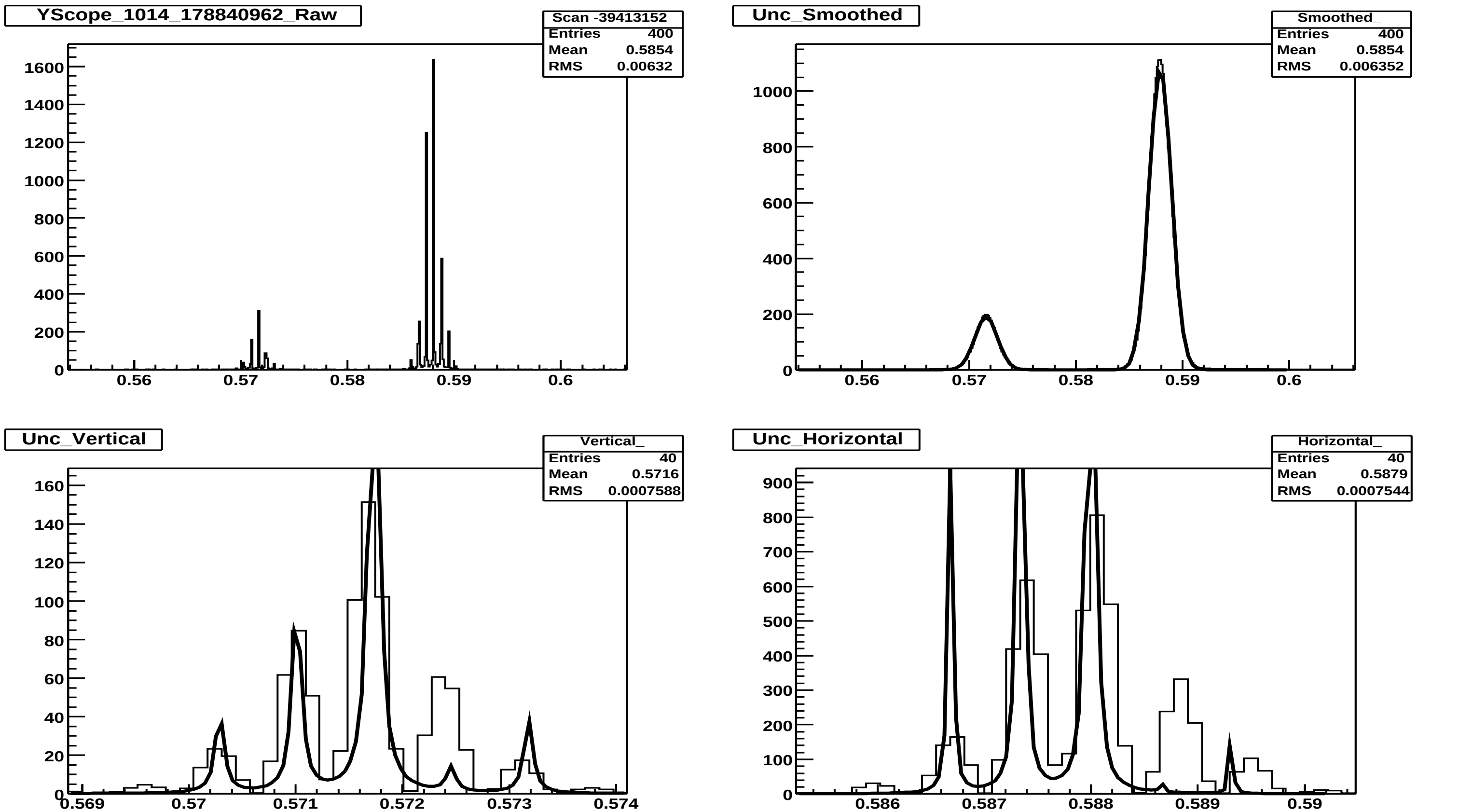


Tunes, V = 0.569115, H = 0.587459, Synch split, H = 0.001812, V = 0.00170, Predicted = 0.00168

Algorithms..Uncoalesced..

- First, Histogram, on a linear Y scale.
- Scale such the noise level (~-80to 70 db) corresponds to few counts per bin.
- Smear (or smooth), on a big scale: every bin content is spread, Gaussian wise, to neighboring bins. This is just a Gaussian convolution or "transform"
- Fit Two Gaussians. This determines the broad value of the Horizontal and Vertical tunes.
- Make two distinct new histograms, one for each region, using the original data.
- Smooth, Cern algorithm, two times.
- Fit with 5 Breit-Wigners, with same widths and same frequency splitting between satellites and main line.

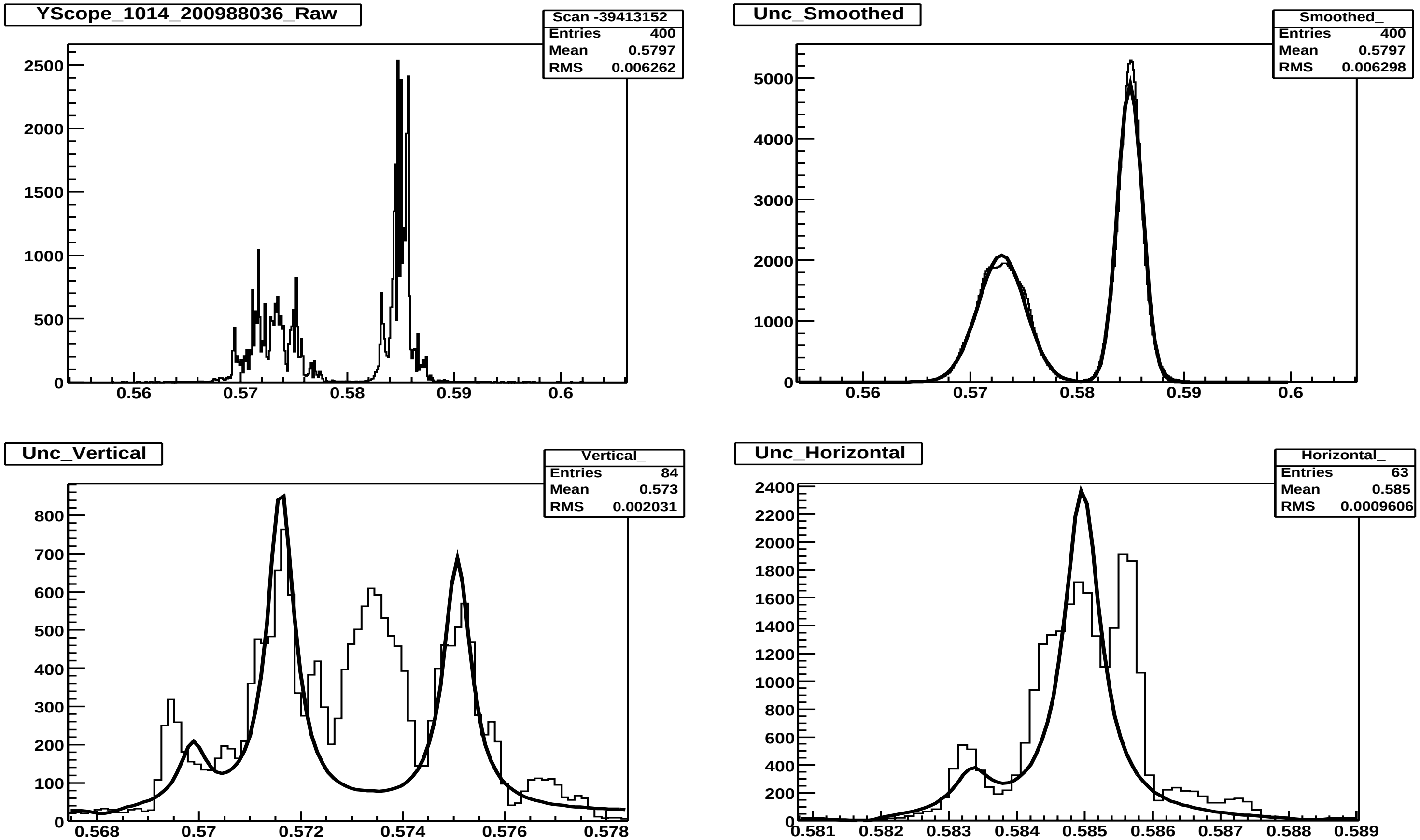
Previous plot at 150, now at 980, same beam..



Tunes, V = 0.571733, H = 0.587999, Synch split, H = 0.0007232, V = 0.000659, Predicted = 0.0065

Back to 150, a bit later..

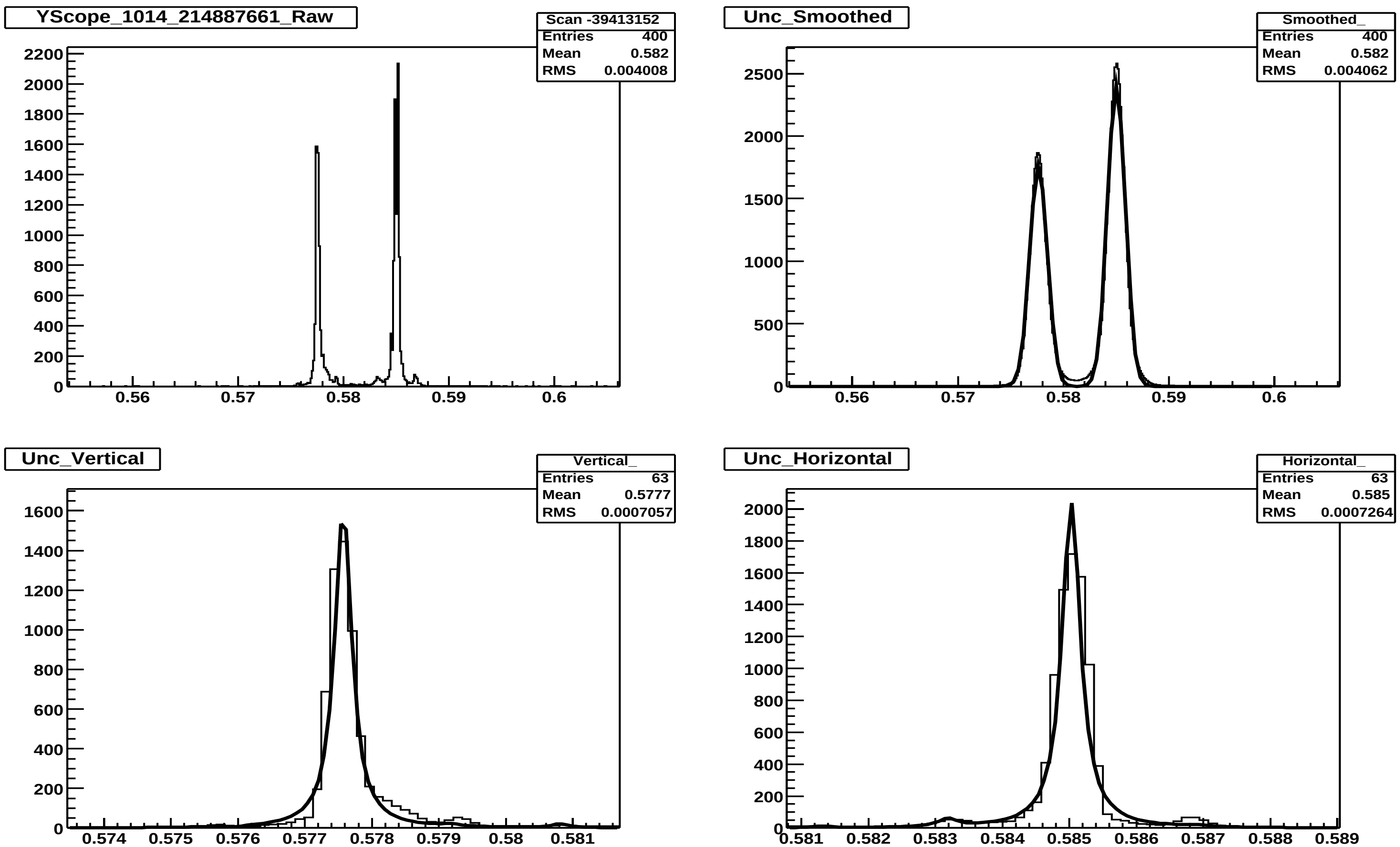
23:04, Dec 11



Despite missed bumps, Synch split, H = 0.0017312, V = 0.0016207, Predicted = 0.00166

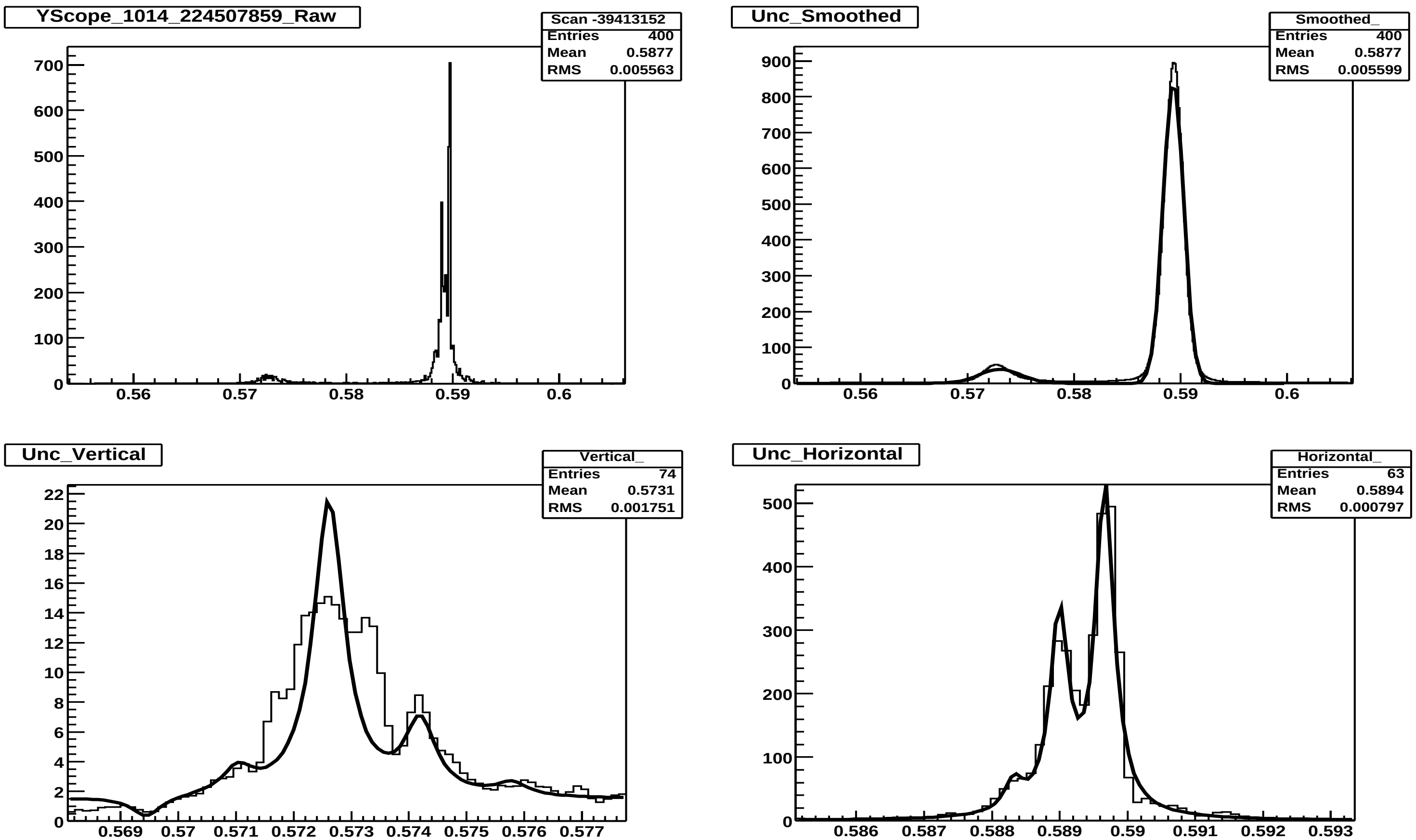
Owl shift…

05:05, Dec 12



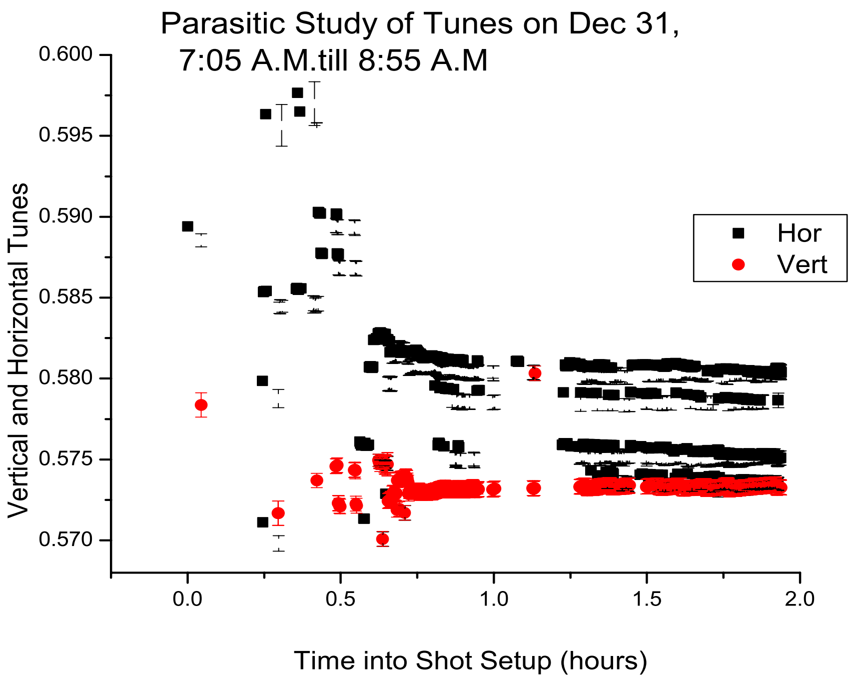
Despite weak bumps, Synch split, V = 0.001799, Predicted = 0.00165

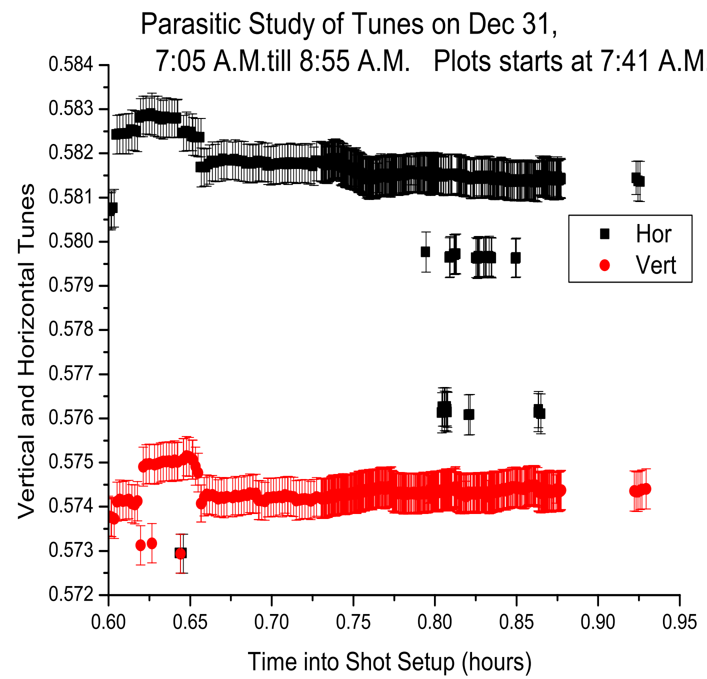
Owl shift…



Despite weak Vertical signal (no tickling, I presume) (10 db above noise), we got a meaningful Vertical tune measurement, Sync-beta (0.00185)

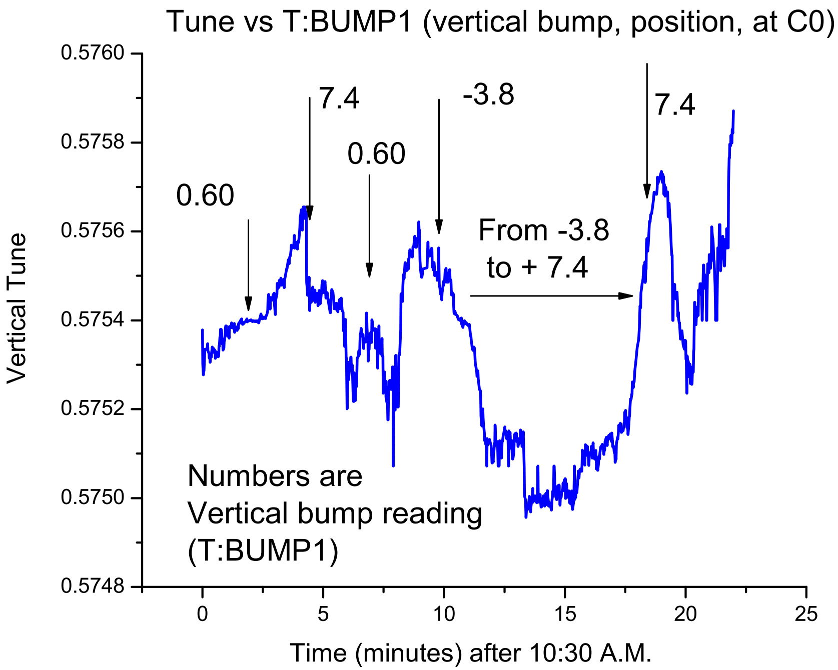
Parasitic Studies in MCR, During regular Shot Setup.

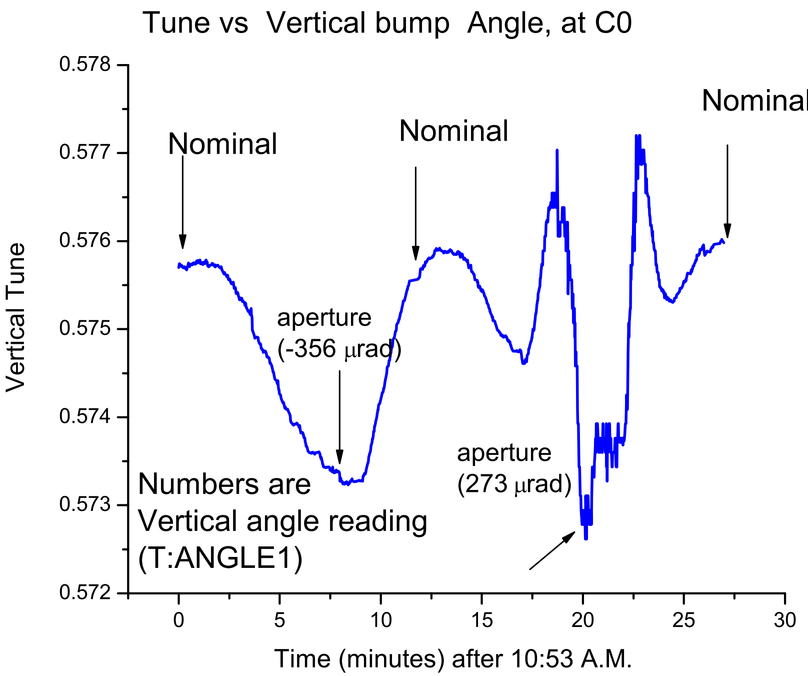




Fitted tunes were "data-logged" (node Inst2) during the shot setup for store 2115 on December 31 2003. During the first 20 min or so, tunes were changed abruptly by operation for standard chromaticity measurement. The tune was completed around ~ 7:40 A.M. For ~ 1.5 hours, the Tev was "left alone" with coasting beam.

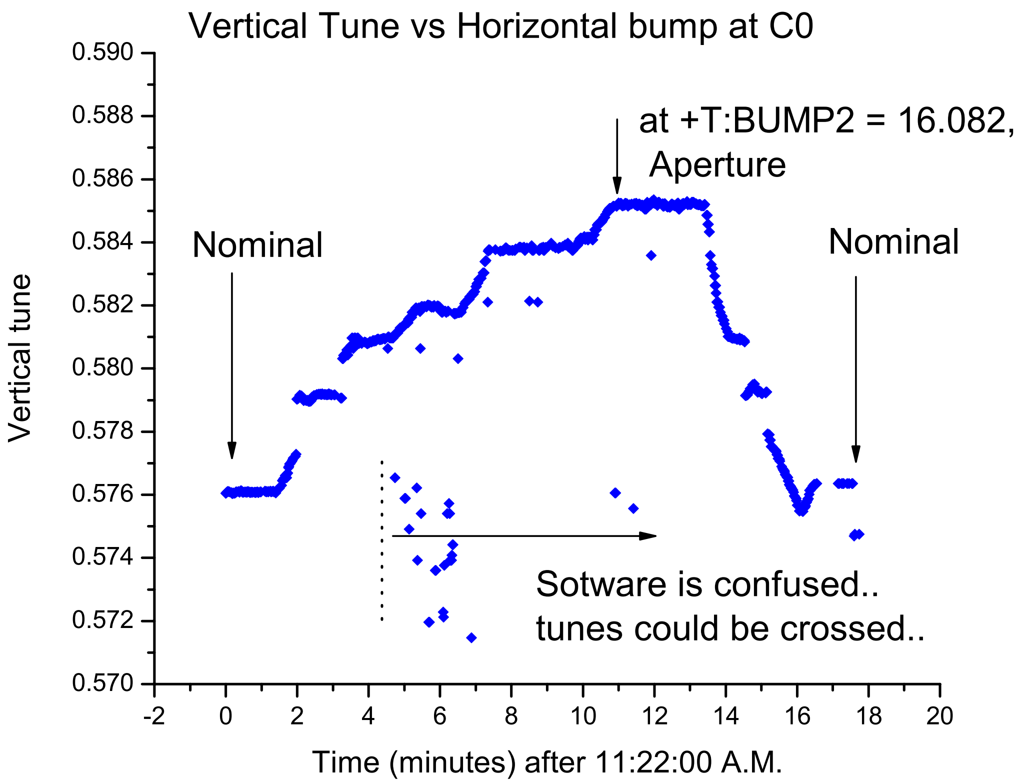
Tunes vs Bump position at C0 (Parasitic)





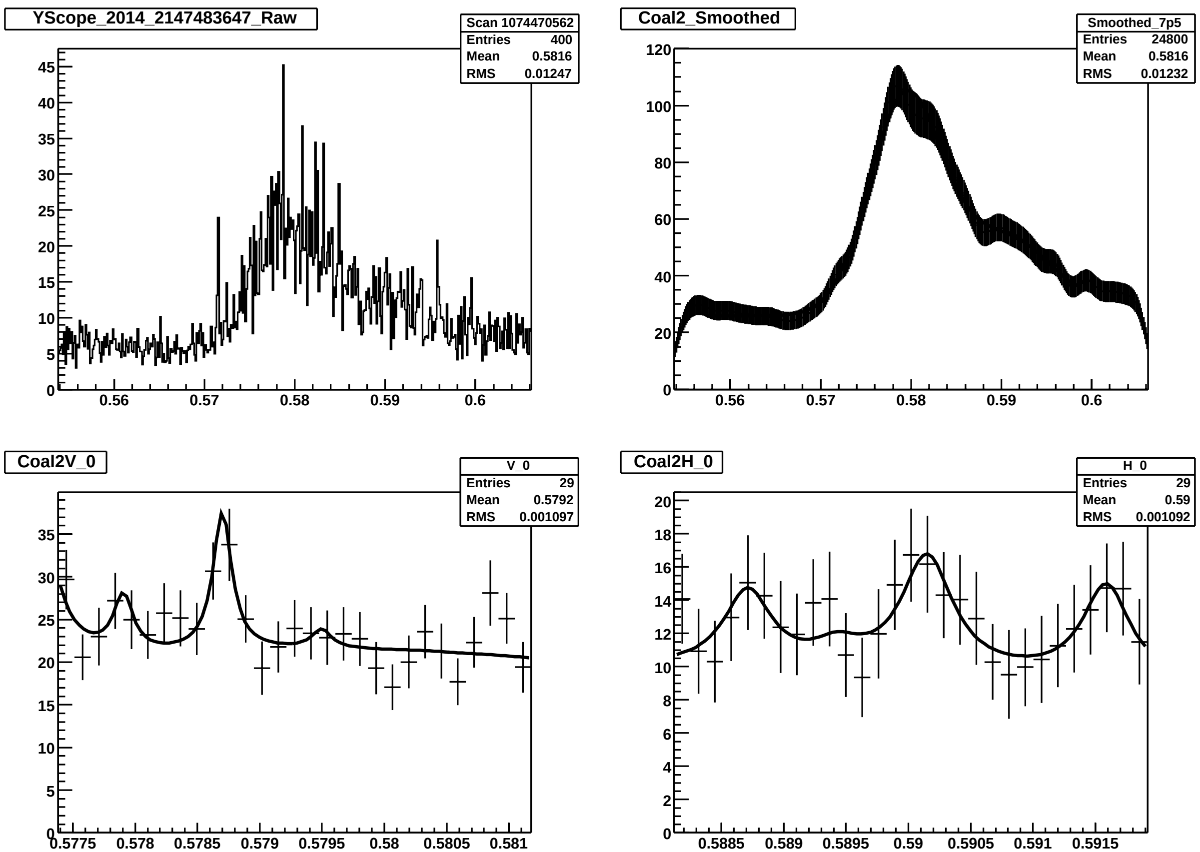
- **The vertical tune is almost insensitive to the vertical position.. Definitely sensitive to angle bump.**

Vertical Tunes vs Bump position at C0 (Parasitic)



- **Very sensitive to horizontal position.**
- **Caveat (again) : tunes did cross wile doing the scan and the software is confused.**

Coalesced, p-bar beams is much harder!

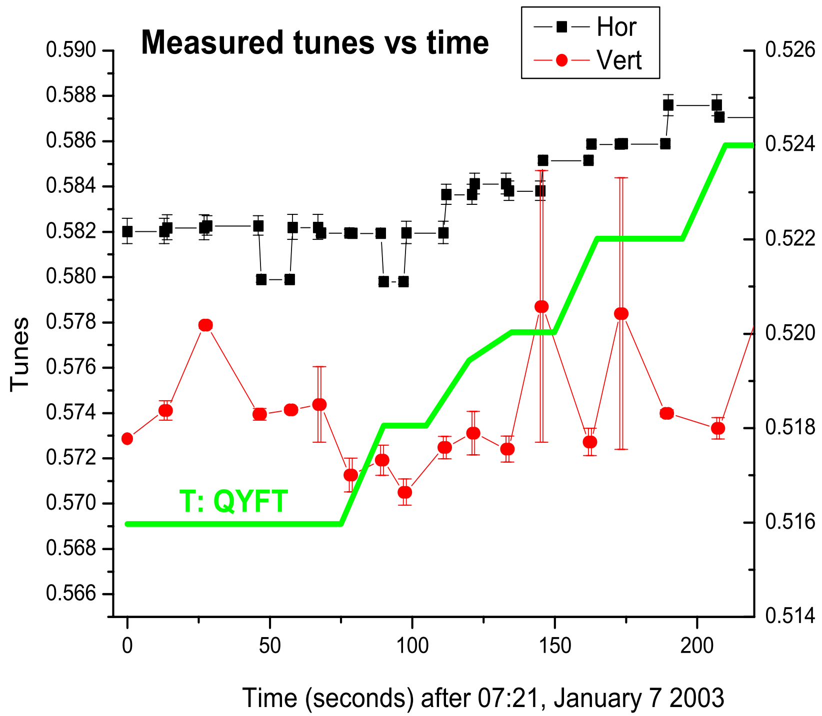


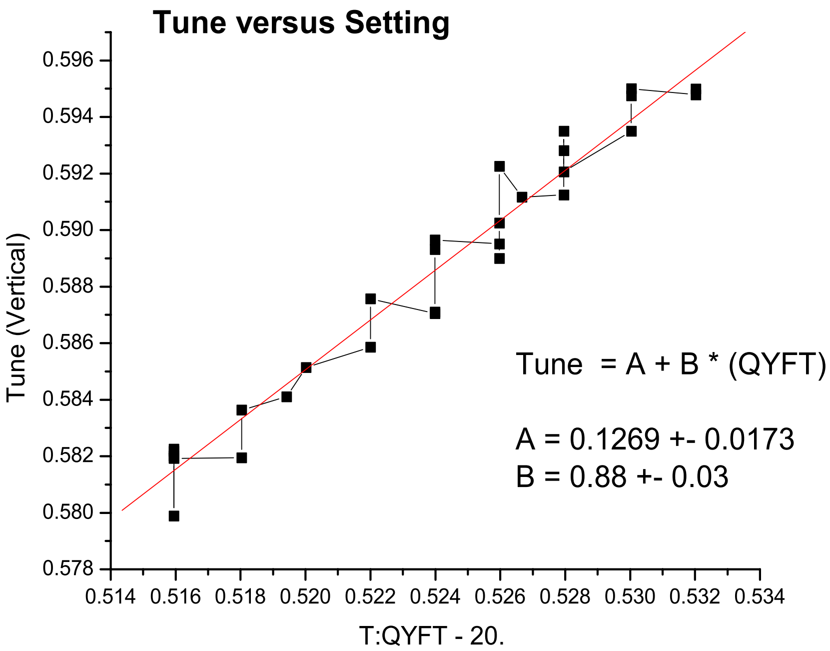
- Data taken on Dec. 16 2002, 11:38 A.M. (store 2078, ~ 2 hours into the store).
- Nothing but noise lines at this point???
- There is more than one tune !
- How do we establish a signal?
- Note : these lines are clearly beam related!

Algorithms..Coalesced Beam(s)

- Overall scenario identical to Uncoalesced. The differences are:
  - 3-Gaussian fits for the broad tunes (instead of 2). The highest tune will be ignored (this needs work, which broad signal to consider the most important relevant one ?)
  - We do these on three different Gaussian-convoluted data sets, with 7.5, 10 and 12.5 bins average, and compute averages between the fitted values. (again, such an algorithm is highly negotiable..)
  - Fit with 5 Breit-Wigners for narrow Synchro-betatron confirmation: the central line (most intense) is allowed to have a different width than the satellites.
- All cases of Coalesced beams are treated identically, although the fitter is aware of the SDA Case name, beam current and of course machine energy.

While Changing T:QYFT,



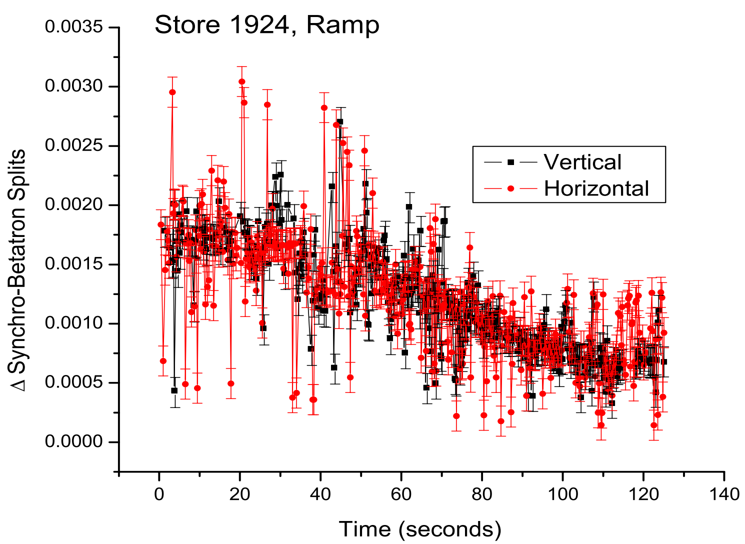


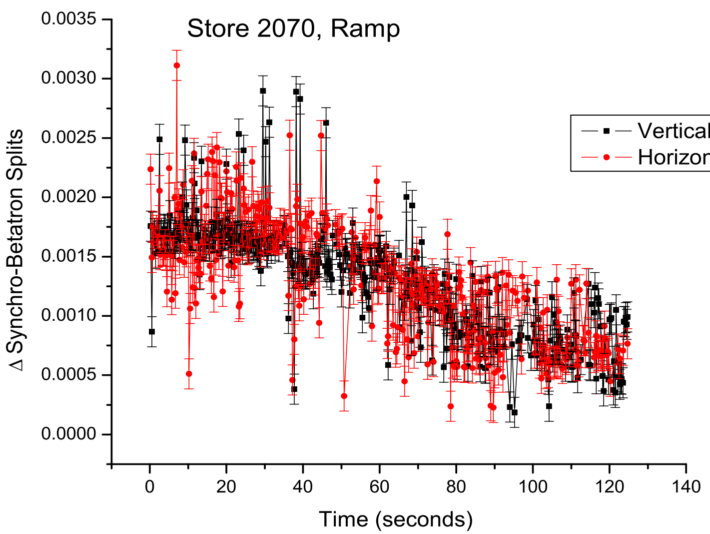
- Datalogging the tunes, setting vs measured. ***Parasitically..***

While Changing T:QYFT, Caveats…

- Taken with Coalesced beam, only two bunches in the machines (~140 e9 and ~57 e9) Evidently, because of finite chromaticity and larger δp/p, it is harder to measured the tunes than with coalesced beam.
- X & Y are flipped! This is because the software, currently, always assigns the *highest* tune to the Horizontal plane. During the scan, the tunes did cross (many times) Two distinct way to fix this: (a good one, and a bad one)
  - Analyze both planes *concurrently* and arbitrate which plane is which based on relative intensities on the lines. To implement this:
    - Minor software upgrade
    - An other Spectrum Analyzer, and corresponding D.A.
    - Twice the CPU power, to keep up..
  - Keep track of the entire history, so that we count the number of tunes crossing. This is difficult, it won't be reliable, especially if the tunes stay very close to each others for long periods of time. Bad idea!.
- There is significant delay (~15 seconds) between a change of T:QYFT and the tune change. This is a "software delay: it takes ~ 5 second for the "fitCoalesced2 and "fitGhost2" class to spit out there results, per scan, on nova (400 Mhz UltraSparc2 CPU). This is too long. Not counting the D.A. latencies, which are different for T:QYFT and T:TUYYBR. => More CPU's , faster D.A.

Do we see these Synchro-Betatron lines? Only on a statistical basis!

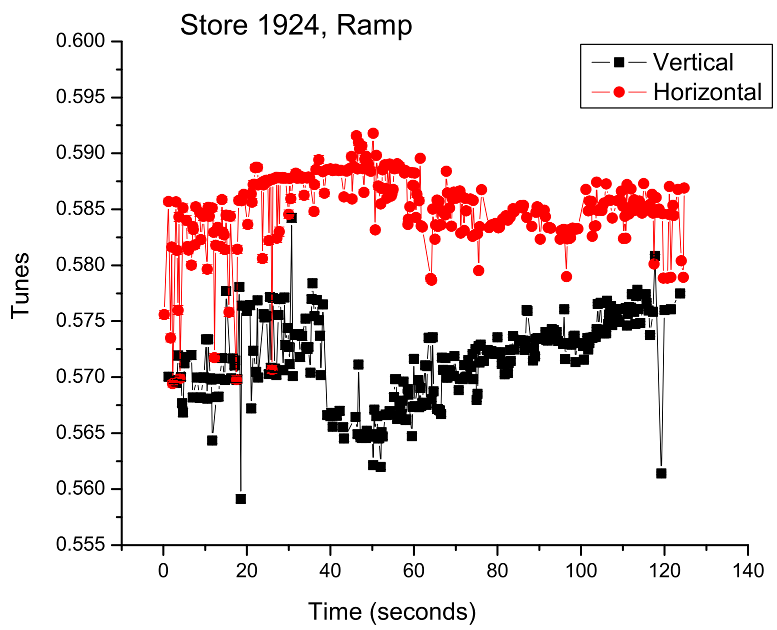


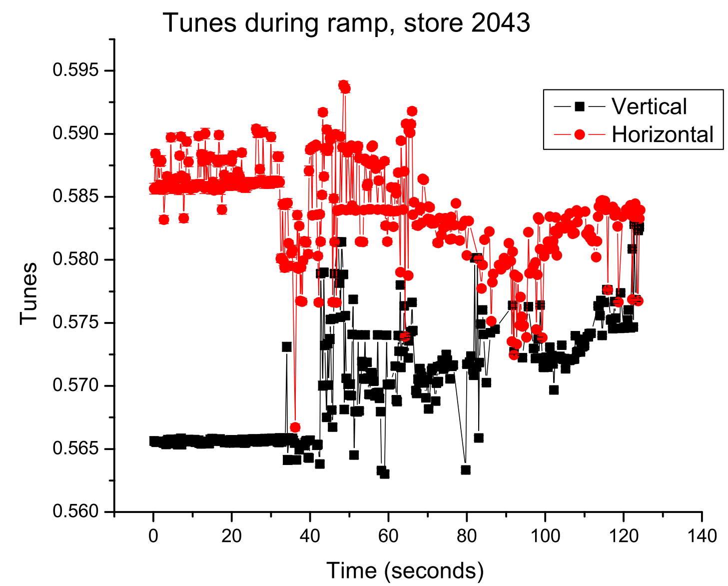


The fitted values for the synchro-betatron line tune split Δsb is plotted versus time during the ramp for store 1924 and 2070. Only results from valid 5 B.W fits (significant amplitudes for the main and at least one satellite line) enter the plot. Fits with a 100 % relative difference between the predicted Δsb and the fitted one are accepted. Yet, a broad band is clearly visible on these plots, centered (within ~10%) on the correct value of Δsb .

If we "seed" the fits with the wrong Δsb value (nominal x 1.25), this band still appears.

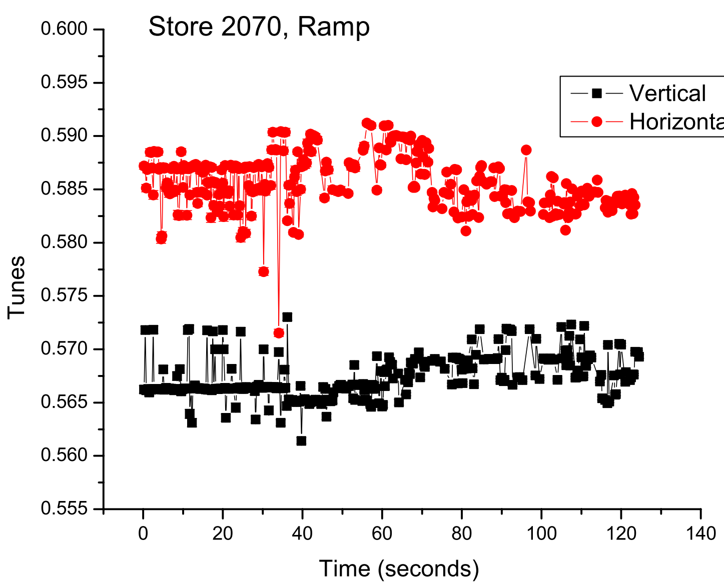
Tunes during the ramp for some stores.

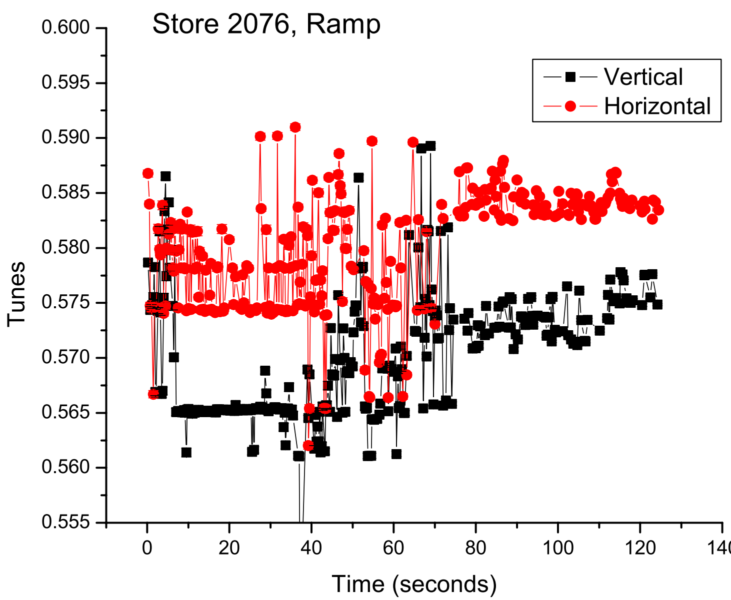




The fitted values for the Δsb tune split has to within 30% of the nominal value. The final tune is set by the sb line closest to the broad tune obtained from the Gaussian Convoluted fits. The algorithm for both planes are identical. Only about 60 to 65 % of the 500 frequency scans from the vsamcr spectrum analyzer have a valid 5BW fit (in at least one plane).

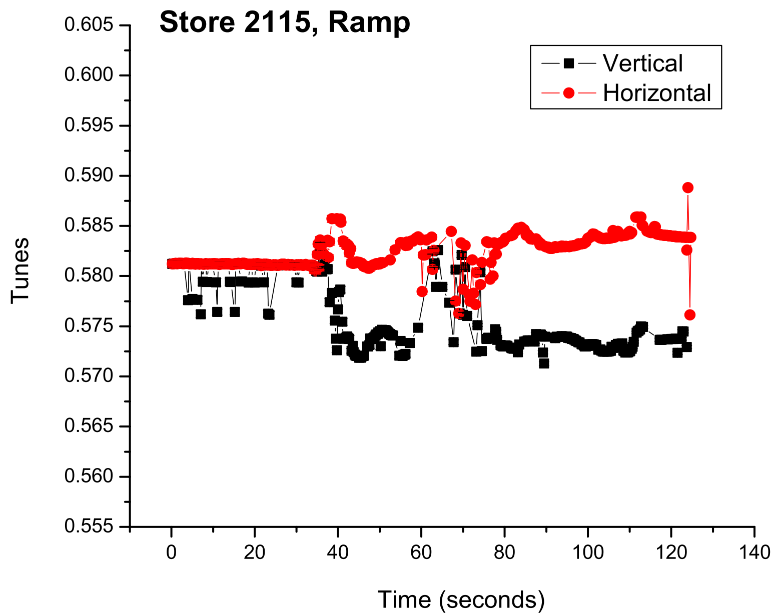
Tunes for the ramp for some stores (2070, 2076)

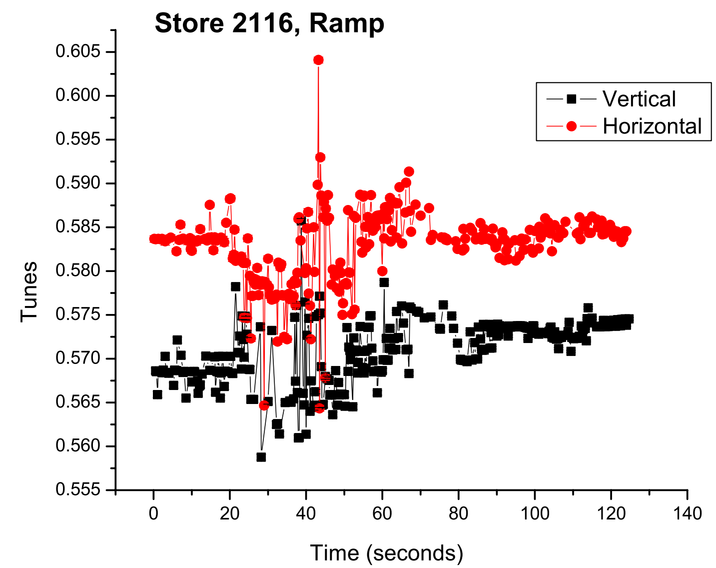




Sudden fluctuations by as much as 0.005. Worse, the vertical tunes are sometimes taken as the horizontal tunes. However, for such beams, the tune spread due to transverse amplitudes, and possibly large chromaticity could in fact explain why some bunches (or excited fraction of some bunches) oscillate at different betatron frequencies.

Tunes for the ramp for some stores. (2115, 2116)



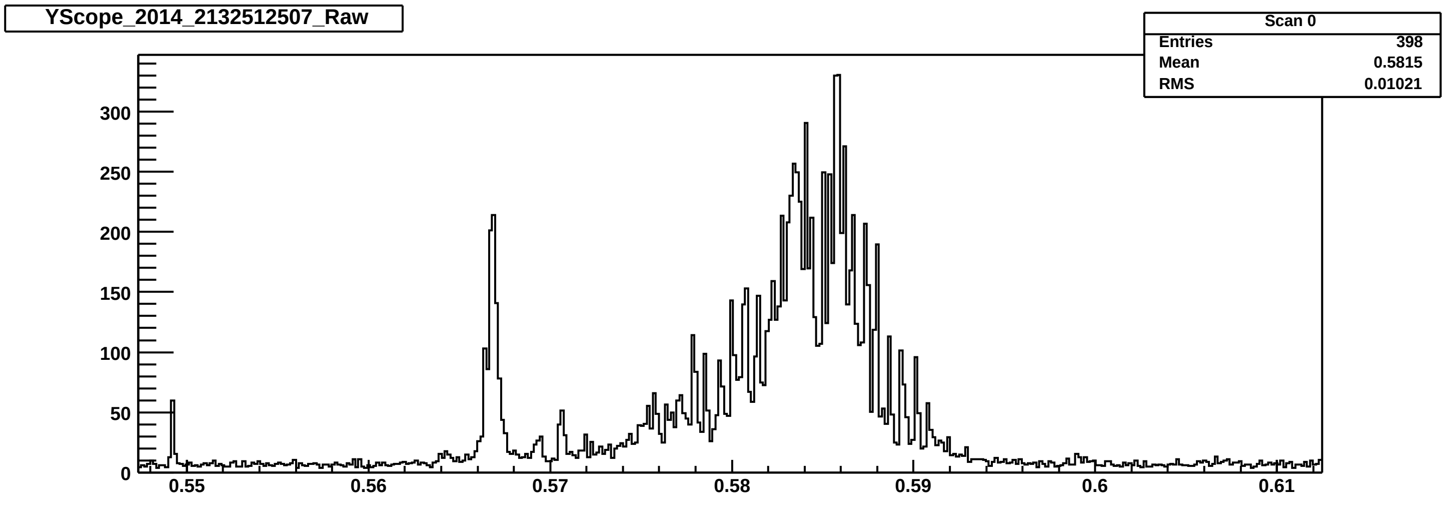


Despite these uncertainties, these tune excursions from nominal values could be of interest. A more systematic studies of such graph, along with step efficiencies and emittance growth measurement should be undertaken.   Note: store 2116, characterized by lower vertical tune at 150, some excitement during the ramp, and with somewhat large proton longitudinal emittance did not last very long (quench at A11 shortly after reaching flat top) Store 2115 was a 2x0 prior to that.

Back to “noise” or “Signal” issue…

- It would be good to make this signal a bit more convincing!
  - “tickle” or “drive the beam resonantly, a bit, to increase signal. How much can we afford without blowing the emittance? Need a dedicated study, Warren Schappert and Dave McGinnis will do this.
  - Or, may in conjunction, we could look at the relative phase betweeen candidate lines.
- So far, only the “scalar” signal analyser has been used (I.e. we use the “vector” signal analyser in scalar mode, for sake of expediency). We could set the vsamcr device in “vector mode”, and collect data. True, we do not have a reference signal, but may be we do not need one, since the evidence for coherent synchro-betatron oscillation is in the relative phase between the main line and the synchrotron line(s).
- Or get a new “vector” signal analyser, so that we do not disrupt the vsamcr device..

“Ghost Lines”.. True noise? A real nuisance!.



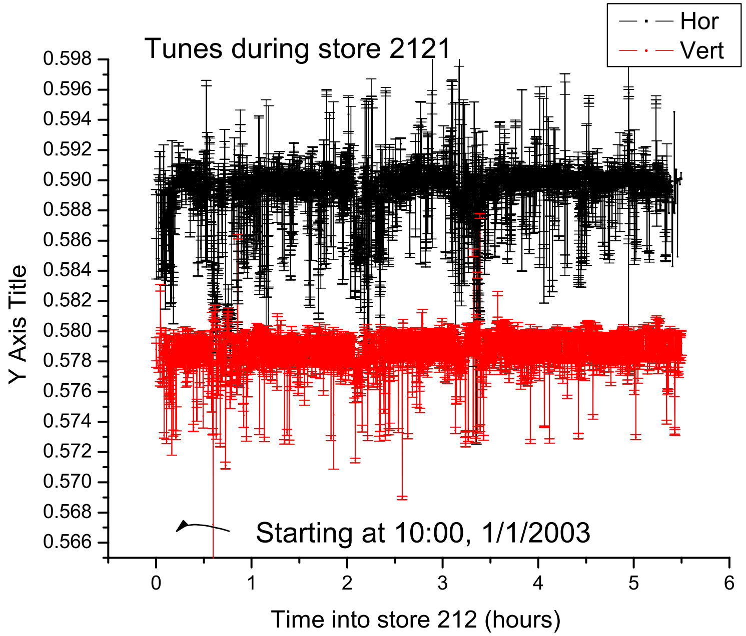
This is a snap shot of the raw spectrum (linear scale), taken during During store 2123 (January 2 2003) Multiple lines are visible. The narrow line at 0.549 is clearly “noise”, as this is outside of the tune map for coasting beam. Moreover, If we suddenly turn the beam off, they disappear! Thus, yes it is noise, but beam related noise.

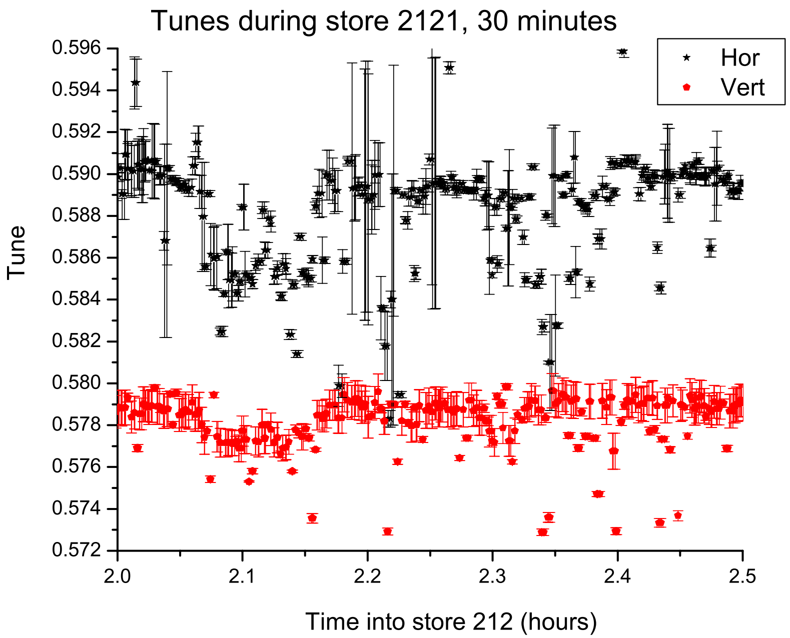
The line at 0.565 is even more troubling, because it is not quite outside the region of interest. It’s often a broader signal, and drifting! And, when it drifts into the betatron lines, it enhances the signal, further evidence for some kind of coupling with the beam.

The Ghost Line (s)…

- Yes, they sometimes drift right into the region of interest…
- In part, this is due to low signal level (only ~ 10 to 20 db above intrumental noise. If we can increase the betatron signal, this would not be such a problem.
- So far, we track only one such line, typically the one around 0.55 The fitting algorithm similar, ("smearing", Gaussian fit, followed by narrow fit). This strategy is not sufficient. An other solution must be found…Since such "noise" is coupled to the beam, somehow, it is not strictly of "instrumental" or "academical" interest, as it could potentially blow the emittance.

Tracking "some tunes" during the store…

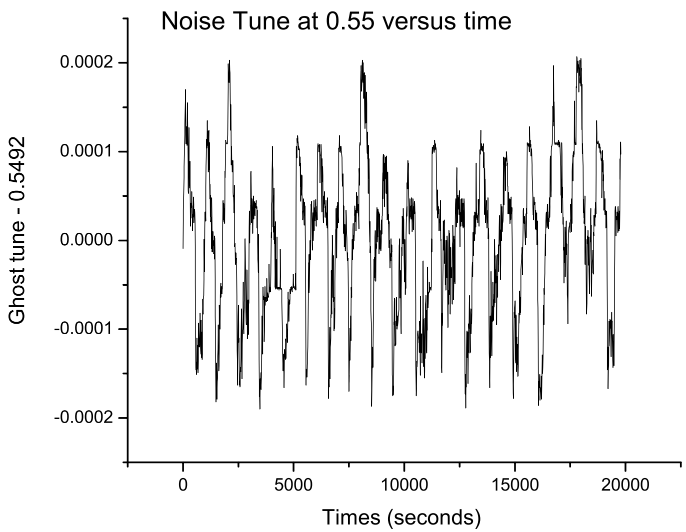


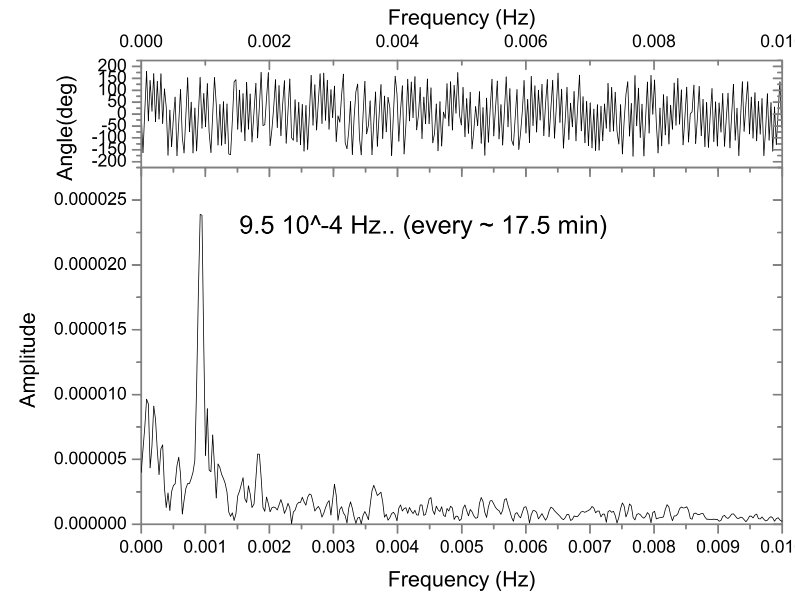


Note: when small errors are assigned, a 5 BW fit has successed, other wise, the error is 0.25 the width of the smeared Gaussians.

The tune fluctuations during the store are smaller than, or about the same as, the tune spread due to finite emittance/chromaticity. The effect of ghost line(s) could also explain such drifts, as while a ghost line goes through the betatron lines, it perturbs the measurements (and, quite possibly, the coherent signal from the beam itself) Correlation of such episodes with beam losses has been not (yet) been established. More work is needed…

Tracking the almost fixed noise line at 0.55…





This mysterious line at 0.55 (26.242 Khz, or an harmonic of this) is remarkably stable. It fluctuate in frequency tune space by ~ 10^-4, not too far from the spectrum Analyzer resolution, with an approximate period of 17.5 min.

The more tricky noise line at 0.565 deserve more attention. Further code needs to be written.

Plans & Schedule: February 03

- Install an other HP3561a, so that of the X and Y proton signal can be readout (almost?) concurrently, and fitted within a few seconds of each other, allowing to lift the XY tune ambiguity. Some software upgrade of fitting package is required (~few man-days) .
- More beam studies, mostly parasitic, some dedicated (for instance, re-calibration of the base tune scale T:QFxx, or other such virtual devices), using the Luciano Picolli Control/D.A. package.
- Possibly, install the Tune Meter software on an other SUN(s) or Linux boxes, so that the load on nova (a development system) can be lifted. Also, manage this "virtual front-end" sofware with CVS.
- Milestone: curves of X and Y tunes "crossing" each others (varying the base tunes). A parametric fit of such curves gives the betatron coupling. (Dedicated study using uncoalesced beam, at any energy).

Plans & Schedule: May 03, I

- Replacement of the two HP3561a and GPIB -> ethernet conversion with a dedicated digitial Tune box, or board(s), with
  - ADC (14 bit or 16 bit)
  - Shark DSP (or Power PC) to do FFTs.
  - Ethernet interface
- *** 4 channels instead of 2 -> Concurrent pbar FFT & spectrum fits
- Installation of the Tune Meter software of a mini-farm connected to ACNET and a dedicated ethernet network connected to the hardware above.
  - 6 PC Processors, one for I/O, 4 for fittings, 1 for spare and tests..
  - Running Linux,(tentatively 500 Mhz Pentiums from old CD farms) equipment. (Cheap !!)
- **** Reduce fitting latency, faster response. Concurrent analysis of X,Y, proton, pbar tune measurement.

Plans & Schedule: -> May 03, II

- Better Communication software, OAC based, for instance.
- Using the Phase information from FFT fits “Joint fits”, Phase and Power..
- Possibly, better graphics via ROOT scripts and procedure to start, stop and control the package.
- More Beam studies.
- Write automated Chrom. And Coupling procedures, with recommended BD/Control software (new (Java) or old (Vax-Sequencers)
- **** Tune Meter fully operational
- TuneTracker Prototype: Based on the BD/Control recommended Feedback software package
- **** Tentative automatic tune stabilization..

Plans & Schedule: “Later”, I

- Possible usage of the new 1.6 Ghz Shottky, instead of the 2.4 Mhz
- Or, conversely, improvement of the analog receiver of the old 21.4 Mhz device..
- Or, find a way to make the 0.565 wandering tune go away..
- “Tickling” specific bunches, allowing tune measurement for a specific bunches. Automate the control software to excite a whole train, sequentially.
- More Beam studies, accurate beam-beam tune shifts measurements.
- **** Accurate Tune Meter for 72 bunches.

Plans & Schedule: “Later”, II

- TuneTracker Implementation & Deployment.
- **** Automated tune stabilization..Possibly during Ramp. Faster store turn around…

Conclusions

- It is possible to express a measurement method into an algorithm that runs on a computer, as fast as one can read from a screen.
- Having the capability of tracking uncoalesced tunes ought to be good, and could, and will (?), be deployed.
- Coalesced beam need further study:
  - A bit of tickle without blowing the emittance ?
  - Noise line
  - Bunch by bunch…
- From prototype to production version: a collaborative effort! I can not do all this alone!

A bit more on history and motivation

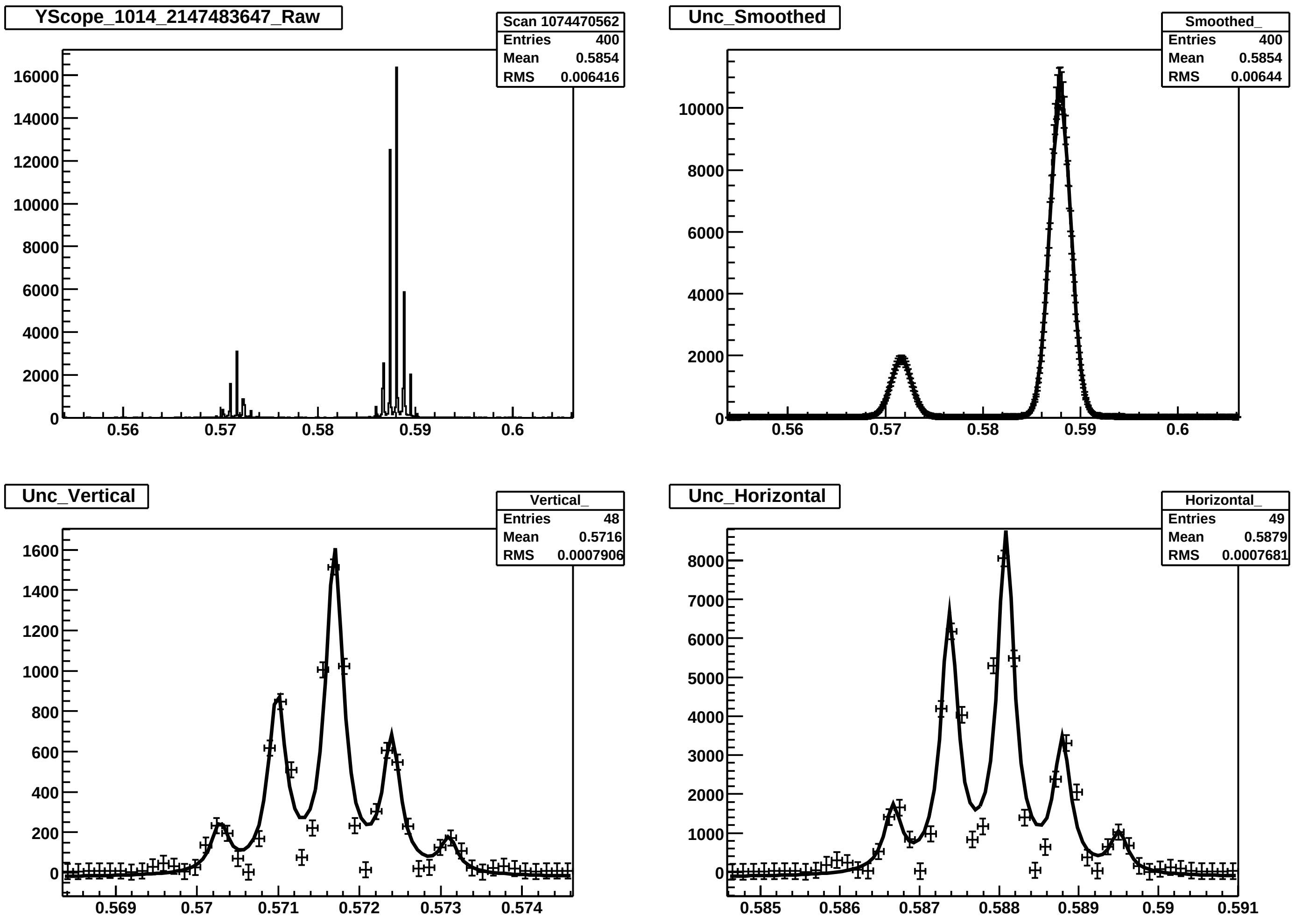
- *The first goal was simply to be able to record the Tevatron tune electronically, and automatically, (instead of having to rely on "human touch" to select the right line), and store the result in SDA.*
- *In addition, I must admit, I got curious to learn how these tunes are measured. I was told that "there is a little bit of black art in choosing the right line". I hope such this bit of secret magic can be described and implemented on a computer!*
- *The project is interesting from the Computer Science aspect: tracking 72 (or more!) tunes (and chromaticity!) independently from each others, for both X and Y planes, and do this "real time", will require high rate D.A., and significant computing power. A parallel implementation will probably be required. In addition, these fits must be intelligent enough to distinguish noise from real signal. The software must be "fault tolerant", must report "the best numbers it can come up with.." Thus, there is a little bit of an "expert system" aspect to it, as the package must be "aware of this black magic". Finally, results must be reported to ACNET.*

Comments of software strategy..

- *C++ has been chosen for the core fitting package, because*
  - *Compiled language, great CPU performance, with efficient use of pointers.*
  - *Language of choice in HEP, we can borrow fancy fitting package. ➔ Using ROOT, and at a later stage the new minimization package written by M. Fischler, M. Paterno, D. Sachs.*
  - *OO, a bit safer, more readable, and cleaner than C*
  - *Dynamical use of data structures prevent us from using plain old Fortran, anyway.*
  - *Java Implementation "postponed" (or rejected…) until we have good fitters with equivalent CPU performance…*
- *Java has been chosen for the reading the spectra because this is what is supported…*

  (We also wanted to learn the DAQ/Control of the futur..)
- *Interface between the C++ and Java code is a straight ASCII file. Relying on Unix I/O subsystem to sort out the locks.. (the read in the C++ code occasionally fails gracefully, as the file is being overwritten. We just try again after sleeping a few hundreds mSec.) A cleaner way would be to implement a UNIX shared memory segment between the Java Native process and the C++ process.. To be done..*
- *Writing the tune numbers to ACNET via XML-RPC, because it is callable from C++, thereby avoiding yet an other clumsy file interface.*

Same data, re-analyzed after algorithm improvements.



Vertical tune = 0.571692, Horizontal = 0.58808